Supplement for: Avoiding test set bias with rank based prediction

Prasad Patil[a], Pierre-Olivier Bachant-Winner[b], Benjamin Haibe-Kains[c], Jeffrey T. Leek[a]

[a]*Department of Biostatistics, Johns Hopkins School of Public Health, Baltimore, MD*
[b]*Institut de Recherches Cliniques de Montréal, Montreal, Quebec, Canada*
[c]*Princess Margaret Cancer Centre, University Health Network, Toronto, Ontario, Canada*

Comparing survival curves

We examined how patient survival is differentiated by the three different subtypers (PAM50, PAM50 unscaled, TSP). This information is replicated in the hazard ratio forest plot in the main text. We find that each predictor differentiates survival across the subtypes in a comparable manner, with large differences between Luminal A patients and all other subtypes. This suggests that the TSP predictor, which is based on only ten genes, performs similarly to the existing PAM50 subtyper while avoiding the test set bias that can befall PAM50.

Modeling with pairwise comparisons

Suppose we have an $M \times N$ matrix of expression values x_ik (with M genes in rows and N samples in columns). Let $z_{ijk} = I\{xik < xjk\}$ and $Z_{ij} = [z_{ij1}, z_{ij2}, , z_{ijN}]$. Z_{ij} is a vector of indicators for the direction of the relationship between features i and j across all subjects. Let $Y = [y_1, , y_N]$ be the vector of outcomes we wish to classify (these can be binary, nominal classes, or continuous values).

The TSP approach [1] for a binary classification problem maximizes the quantity $s_{ijk} = |E[z_{ijk}|y_k = 1] - E[z_{ijk}|y_k = 0]|$ by estimating each expectation with the average. The original approach classified based on the one highest-scoring pair, while later approaches such as k-TSP have extended to using the k top-scoring pairs with methods for selection of k [2]. An alternative approach to viewing the problem is to model the outcome conditional on the pairwise indicator value. In other words we write down a model for $E[y_k|z_{ijk}; i, j = 1, \ldots, M]$. In principle, it is possible to use logistic regression or any more complicated machine learning algorithm [3] to build a predictor for the outcome on the basis of the pairwise comparisons.

By reversing the conditioning argument we can create a simple extension to the top-scoring pairs idea for multi-class classification problems. We build a model for $Pr(y_k = \text{class } j|y_k, z_{ijk}; i, j = 1, \ldots, M)$. We can create a score that is equal to $s_{ijk} = Pr(y = \text{correct}$

Email addresses: bhaibeka@uhnresearch.ca (Benjamin Haibe-Kains), jtleek@gmail.com (Jeffrey T. Leek)

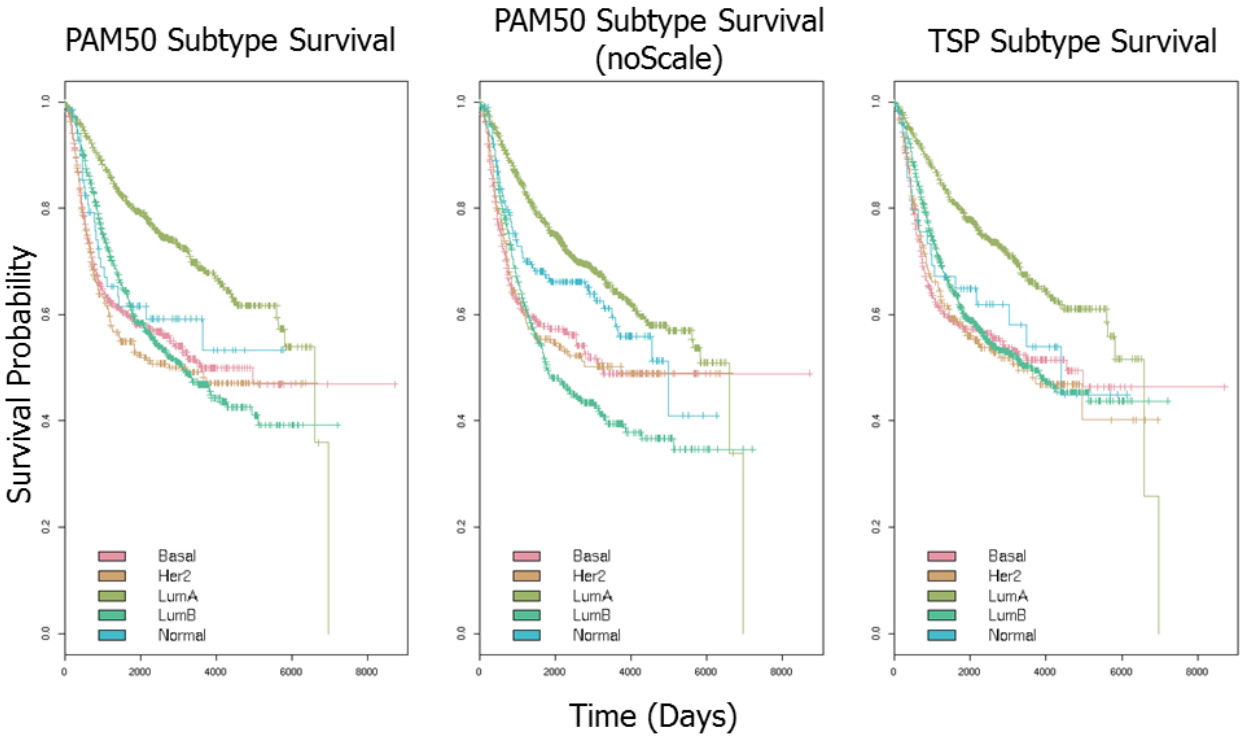

Figure 1: **Comparison of survival differentiation by subtype for each of the three examined prediction approaches.** The reference category in each case is Luminal A, and we see a similar visual pattern of differentiation across all three methods. This is supplemented by the hazard ratios presented in the main text forest plot, which support that each model is differentiating survival in a similar manner.

class $|z_{ijk})$ for each i, j combination. Our modeling approach then proceeds to select the pair that optimizes s_{ijk} in a greedy fashion.

One difficulty is that the score s_{ijk} is calculated independently for each feature pair, so it is possible that the top k pairs may repeatedly differentiate the same classes well while neglecting other classes of a multi-class outcome. To rectify this, we applied a conditional approach to building decision trees, which prioritizes accuracy gain conditional on the features already in the model when choosing a new feature. Specifically we used the following algorithm:

Algorithm 1

1. Separate the data into a building and evaluation set.
2. Fit the model $s_{ijk} = Pr(y = \text{correct class}|z_{ijk})$ for each pair i, j in the building set and calculate misclassification error $Pr(y = \hat{\text{correct class}})$ in the evaluation set.
3. Let i^1, j^1 be the index of the pair with the minimum misclassification error. Fit the model $Pr(y = \text{correct class}|z_{i^1 j^1 k})$.
4. Repeat until the number of pairs is equal to the maximum total pairs
   (a) At pair ℓ fit models for each pair i, j conditional on all previously selected pairs $Pr(y = \text{correct class}|\{z_{i^t j^t k}\}_{t<\ell}, z_{ijk})$
   (b) Add pair i^ℓ, j^ℓ to the model if it reduces the misclassification error, if not stop.

For the work presented here, our main goal was to improve on the existing PAM50 subtype classifier which requires 50 genes to predict the breast cancer subtype for each sample. So our list of potential features included only pairs where both genes were drawn from the original PAM50 signature. We build the model for the multi-class probability based on classification and regression trees [4] using the rpart package in R [5].

"Chimeric" Data Modeling

To reduce platform-specific effects we combined records from different platforms and training a prediction model on the resulting "chimeric" dataset could improve overall cross-platform accuracy. We combined datasets by taking a fixed number of sample records from every platform that we wished to represent (such that the resulting data had equal contribution from each platform) and produced a dataset merging all records and containing the intersection of probes available on each platform. Importantly, we did not process the data in any way when we created the merged dataset.

We explored the possibility of probe effect mitigation by training on a chimeric dataset. The hypothesis here was that training on a mixed dataset would ignore features that did not have the same direction and magnitude of a relationship across all included platforms. We first ran a simulation where we built 100 models each on Affymetrix hgu133plus2 (GSE5460), Agilent G4110A (ISDB10845), and Illumina HT12-V3.0 (ISDB10278), and the chimeric mixture of the three. We then examined which genes were chosen most often for pairs across the platforms (heatmap in **Figure 2**). We paid special attention to the gene BIRC5, which

was chosen often in Illumina but was hardly chosen in any other platform or in the mixture. We also ran a smaller, 25-model simulation to examine which pairs were chosen most often, and indeed BIRC5<KRT14 is the pair chosen most often for the Illumina dataset (indicating that it was most predictive of the outcome). Looking at how each pair differentiates the outcome across platforms (**Figure 2**), we see that this particular pair works completely differently on the other two platforms, leading to it not being chosen for the chimera model. We see that there are other such broken pairs in the Affymetrix and Agilent platforms that are also not considered in the chimera. Since we are using only three datasets, however, it is unclear if this pair broken in all datasets but Illumina or just in the two Affy/Agilent sets we are comparing with. These results help establish why the chimeric model is more robust across platforms, as it averages out a strong, single-platform probe effect and chooses pairs that behave consistently across platforms.

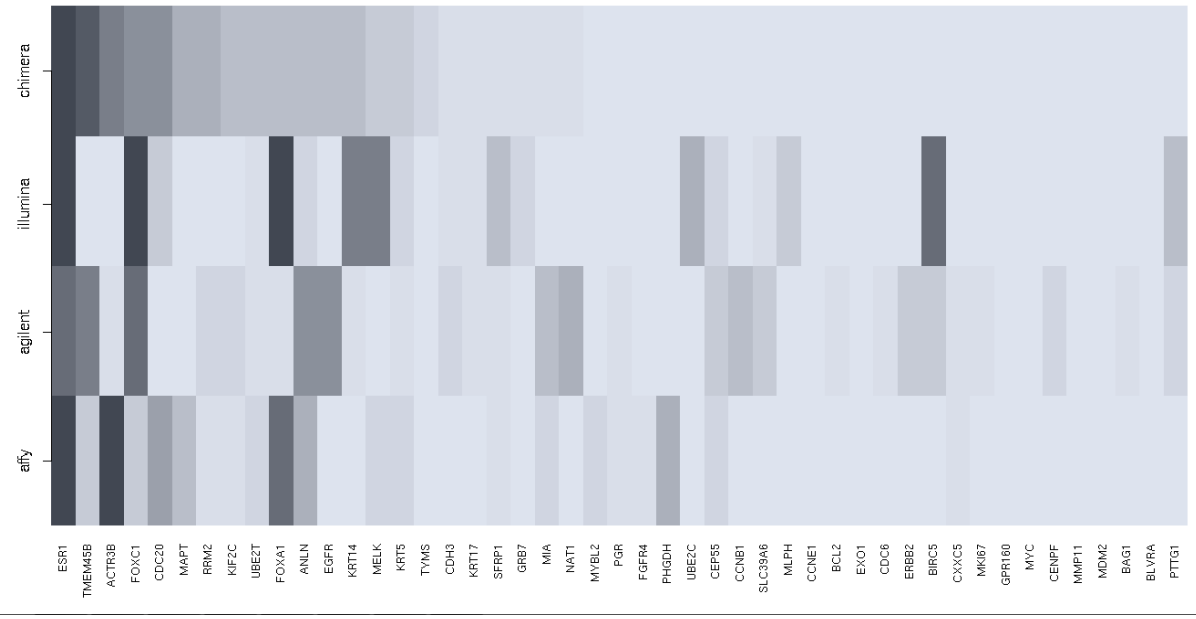

Figure 2: After 100 simulated model fits using each platform listed, we noted which genes were chosen each time and with what frequency overall. The heatmap ranges from black to white, with a darcker rectangle indicating that the gene was chosen more often. The heatmap is sorted by genes chosen in the chimeric model. We see general agreement (i.e., genes chosen often in single-platform models tend to be popular in the chimeric model). However, there are some genes (BIRC5) that are chosen relatively often by one platform but not considered by chimera, and vice-versa (MAPT)

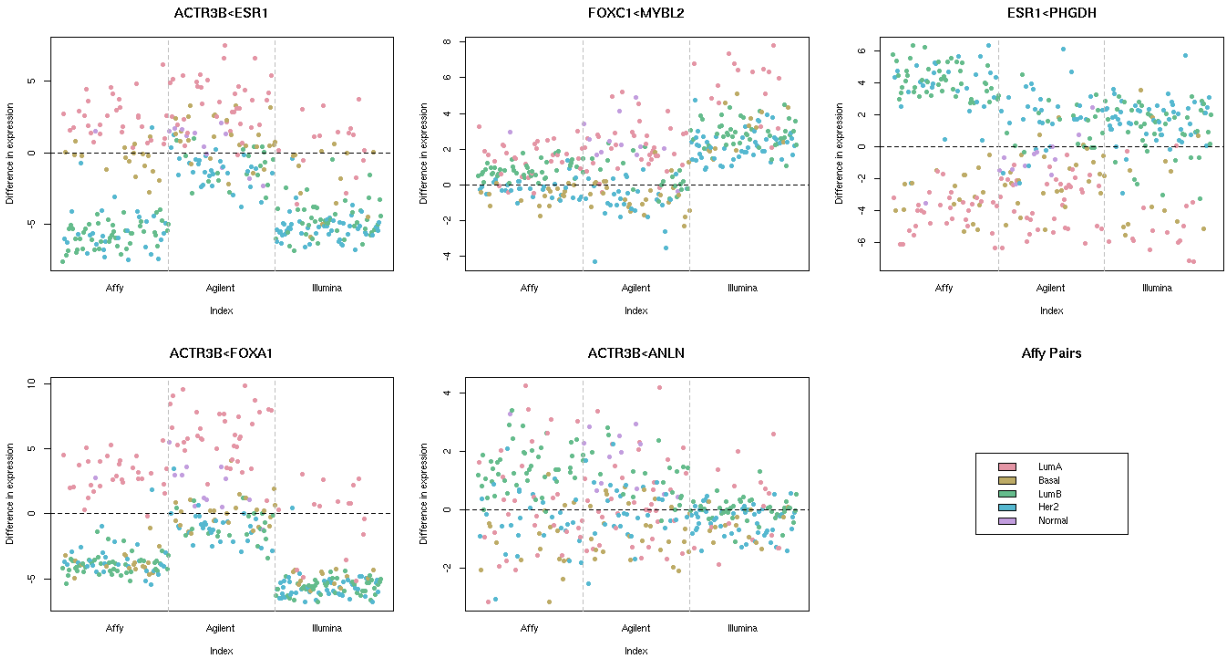
ACTR3B<ESR1
FOXC1<MYBL2
ESR1<PHGDH
ACTR3B<FOXA1
ACTR3B<ANLN
Affy Pairs
Difference in expression
Affy
Agilent
Illumina
Index
LumA
Basal
LumB
Her2
Normal

Figure 2a

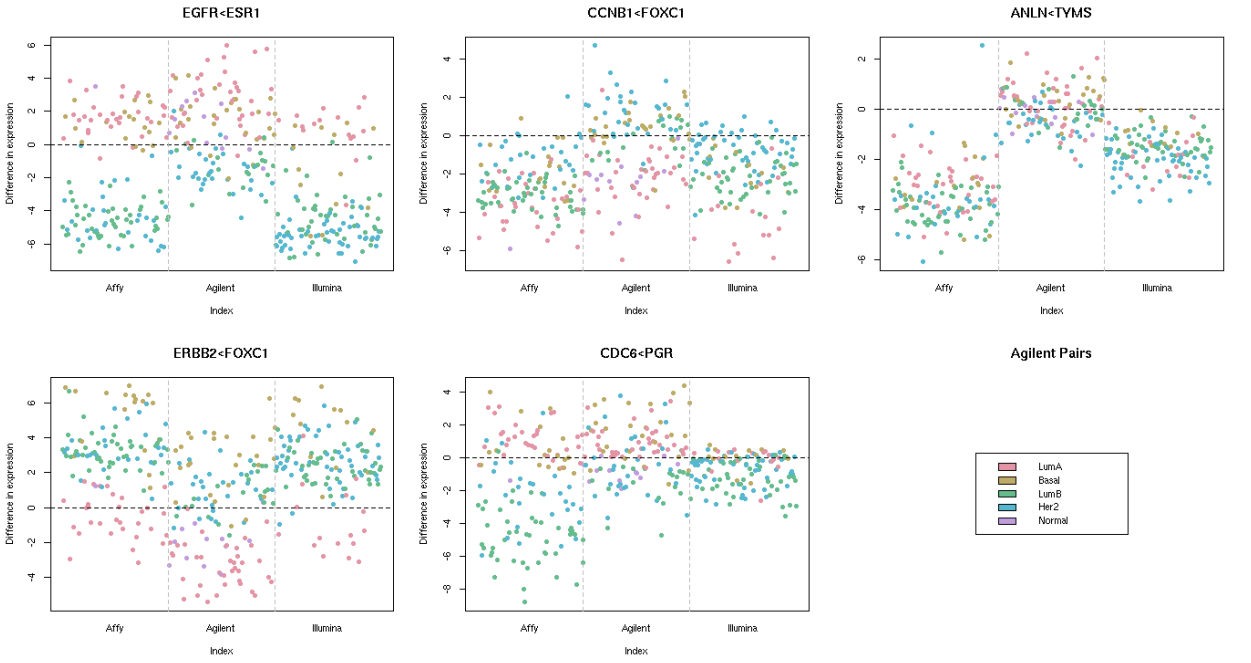

Figure 2b

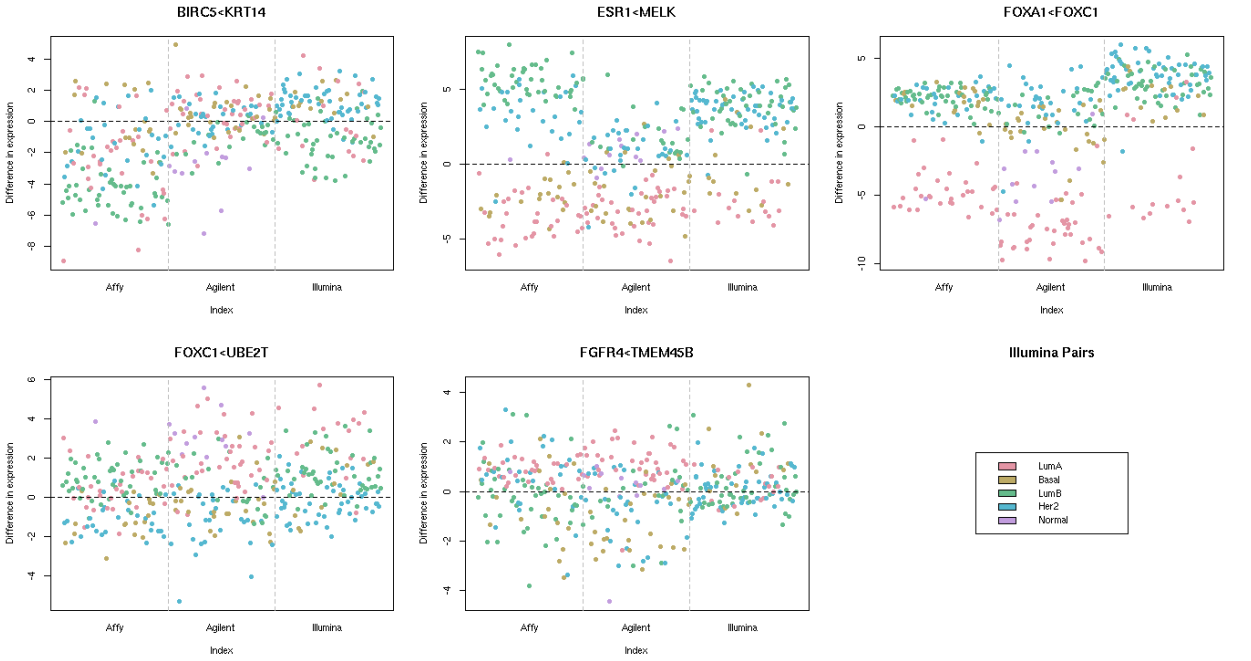

Figure 2c

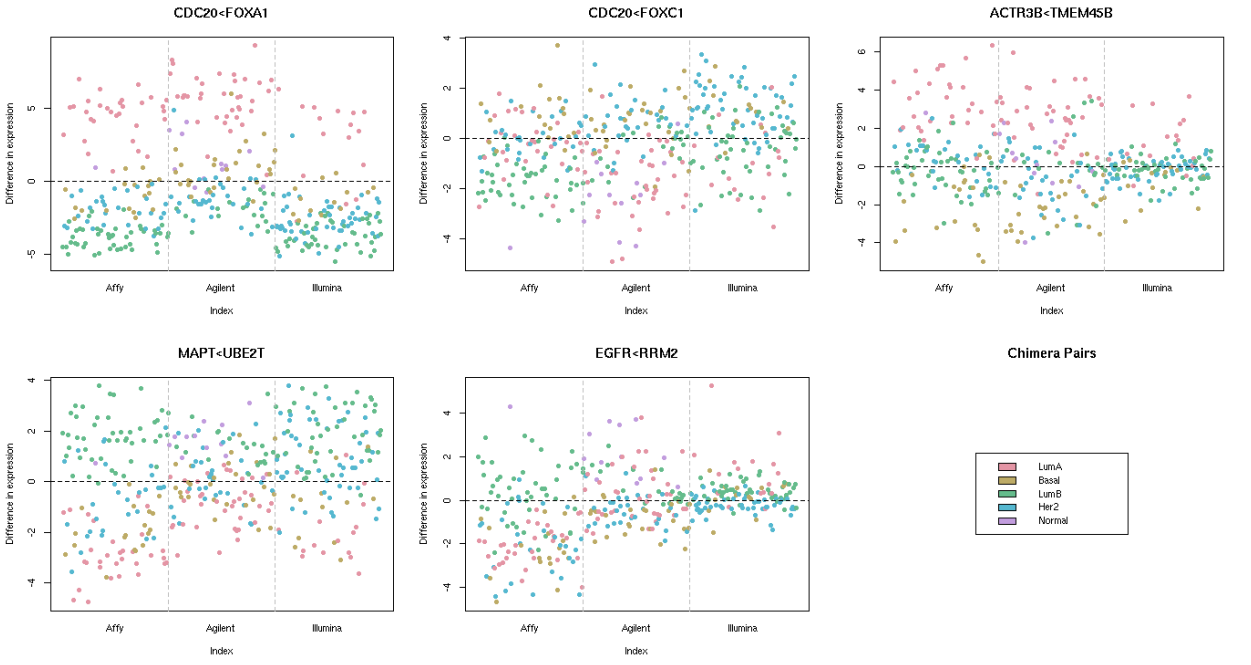

Figure 2d

Figure 3: The above figures show how each pair in each model classifies the molecular subtype labels. We plot the difference in the gene expression for each sample in the chimeric data. If a pair is a good cross-platform differentiator, we should see the same colors above and below the dotted zero line across all three platforms. In Figure 10c, for example, we see that the top pair chosen in Illumina, BIRC5<KRT14, does not exhibit the same behavior across platforms.

Model tree for handling missing genes

One issue with cross-platform comparisons is that some genes that appear on one platform do not appear on the others. In this case, there is systematic missingness across platforms in the genes that are present. We therefore created an approach for dealing with missingness that did not require additional genes to be used in the signature. This is an important constraint, as replacement genes can dramatically increase the size of genomic signatures. Since our focus was to build small predictors (with number of feature pairs $\sim$5), we settled on a model tree approach.

First we fixed the maximum signature size to be $L = 5$ pairs for all classifiers. We chose 5 pairs because we wanted our classifiers to comprise fewer than 10 genes and thus be amenable to economically assayed on large populations.Next, we fit every nested submodel possible among the $L = 5$ selected features. In other words, we retrained classifiers to use any possible combination of the five features for prediction. The result was $2^N - 1$ total trained models. We only retain those models using at least two pairs and we store these in a tree with labels indicating which of the L pairs were used to build the model in a given leaf. For example, if we had 5 pairs, the model under the leaf labeled $\{1, 0, 1, 1, 0\}$ would have used the 1^{st}, 3^{rd}, and 4^{th} pairs. When attempting to predict on the target dataset, we use the model with the maximum number of available features.

The effect of different missing features can be quantified

To examine the effect of missing probes, we artificially removed all possible subsets of the eight genes that comprise the five pairs in the chimeric TSP model from the Affymetrix test dataset (the same data used for the stability tests in the previous section). We then compared predictions produced on these subsets by the corresponding subset model to the original five-pair predictions. Results appear in **Figure 4**. We see that pairs 2 and 3 seem to be strongly associated with predictive accuracy, as the range of their concordance with the original predictions over all models that contain either pair 2 or 3 has a higher median than the rest of the pairs. We believe that the model tree approach can help investigators determine what number of pairs and which specific pairs are most necessary for maintaining predictive accuracy. Many existing models will propose a set of primary predictors and multiple backup sets should any of the primary predictors be missing. The model tree offers an alternative that may help keep the cost of a genetic test down because we would not need to sequence any backup genes. If our model can predict breast cancer subtype reasonably well with 10 genes, we can create chip that measures just those 10 genes and forgo the additional 20, resulting in a cheaper genetic test.

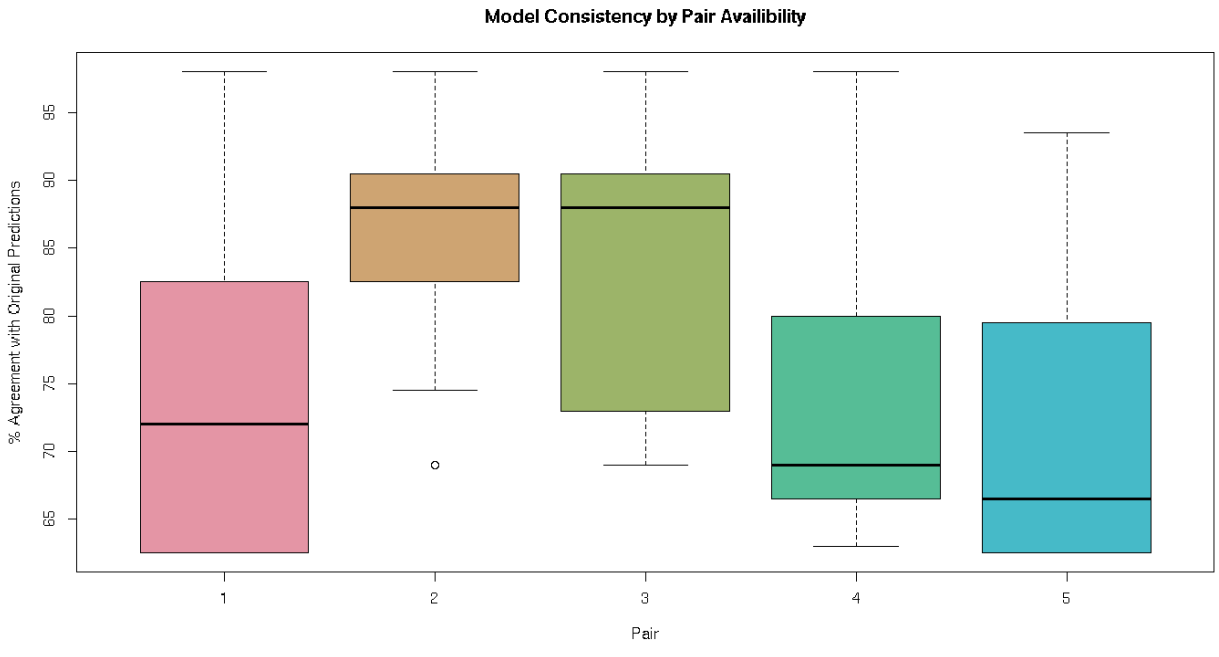

Figure 4: From the chimeric signature, we removed all combinations of the eight genes included and compared the resulting predictions from each submodel to the original, five-pair predictions. Each boxplot shows the range of retention of all models when the specific pair is in the model, i.e., the boxplot for Pair 1 shows how well each model containing Pair 1 agrees with the original predictions. We are able to identify certain pairs (Pair 2 and 3) that seem more necessary for maintaining the predictions.

References

[1] D. Geman, C. d'Avignon, D. Q. Naiman, R. L. Winslow, Classifying gene expression profiles from pairwise mRNA comparisons, Stat Appl Genet Mol Biol 3 (2004) Article19.

[2] J. A. Eddy, J. Sung, D. Geman, N. D. Price, Relative expression analysis for molecular cancer diagnosis and prognosis, Technology in cancer research & treatment 9 (2) (2010) 149.

[3] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The elements of statistical learning, Vol. 2, Springer, 2009.

[4] L. B. J. F. R. Olshen, C. J. Stone, Classification and regression trees, Wadsworth International Group.

[5] T. Therneau, B. Atkinson, B. Ripley, rpart: Recursive Partitioning, r package version 4.1-4 (2013). URL `http://CRAN.R-project.org/package=rpart`